

Adatbázis alapú rendszerek

2023-2024/2
IB152L-10

Quiz

Készítette:
Kávai Illés
Kaszás Zsolt József
Velicán-Benkő Attila

Munka felosztása

Közösen dolgozunk a feladatok megoldásán.

Értékelési mód:

A csoport tagjai közösen dolgoznak, minden tag ugyanannyi pontot kap.

Feladat szöveges leírása

A weboldal egy quiz játék. Amelyben a felhasználók saját szobát készíthetnek. Beállíthatnak saját kérdéseket azokra válaszokat. Ezeket a szobákhoz más játékosok csatlakozhatnak és megnézhetik, hogy mennyi kérdésre tudnak válaszolni az adott szobában.

A weboldal, php-ban készül. Az Oracle 12.0.0.0.1 adatbázist és a SQLDevelopert használjuk az adatbázis létrehozásához és kezeléséhez. A dokumentációhoz a word programot választottuk. Az alkalmazás fejlesztése Visual studio code/ Webstorm fejlesztői környezetben történik. Diagrammok készítésére a következőket használjuk: Draw.io, ClickCharts.

Az adatbázis tartalmazza az hét fő adattáblát: felhasználó, eredmény, kérdés, téma, válasz, *szoba*, *szoba_kérdesei*. Az adattáblákat az adatbázis kezeli és frissíti az alkalmazás által.

Az alkalmazás a belépő képernyővel indul, ahol a felhasználó beléphet a felhasználói fiókjába. Ha a felhasználó még nem regisztrált, akkor a regisztrációs űrlapot kell kitöltenie. A sikeres bejelentkezés után a felhasználó a főmenübe kerül, ahol lehetőség van, szobát létrehozni, szobába csatlakozni, kijelentkezni, profil megjelenítésre.

A szobába való belépést követően lehetőség van kitölteni a quizt. A játékban mindig csak egy kérdés aktív, és a válaszok mindig csak a kérdésre vonatkoznak.

A játékmenet során a játékosok a quiz kitöltőivel versenyeznek, eredményük eltárolásra kerül az adatbázisban. A játék során a játékosok különböző kérdésekre válaszolnak. Az adott kérdésekhez tartozó válaszokat az adattáblából olvassa be a program, és a válaszok helyességét is az adatbázisban tárolja.

Követelménykatalógus

Cél: Az alkalmazás célja, hogy interaktív és szórakoztató módon lehetőséget biztosítson a felhasználóknak szobákban vagy nyilvános helyeken való versengésre, ahol különböző témákhoz kapcsolódó kérdésekre válaszolhatnak.

Funkcionális követelmények:

- A játékosoknak lehetősége kell, hogy legyen a alkalmazásba való belépésre egy felhasználónév és jelszó segítségével.
- Regisztráció
- A játékosoknak lehetősége kell, hogy legyen a játékszobákhoz való csatlakozásra
- A játékosoknak lehetősége kell, hogy legyen új játékszobák létrehozására, amelyeknek nevét és jelszavát meg kell adniuk.

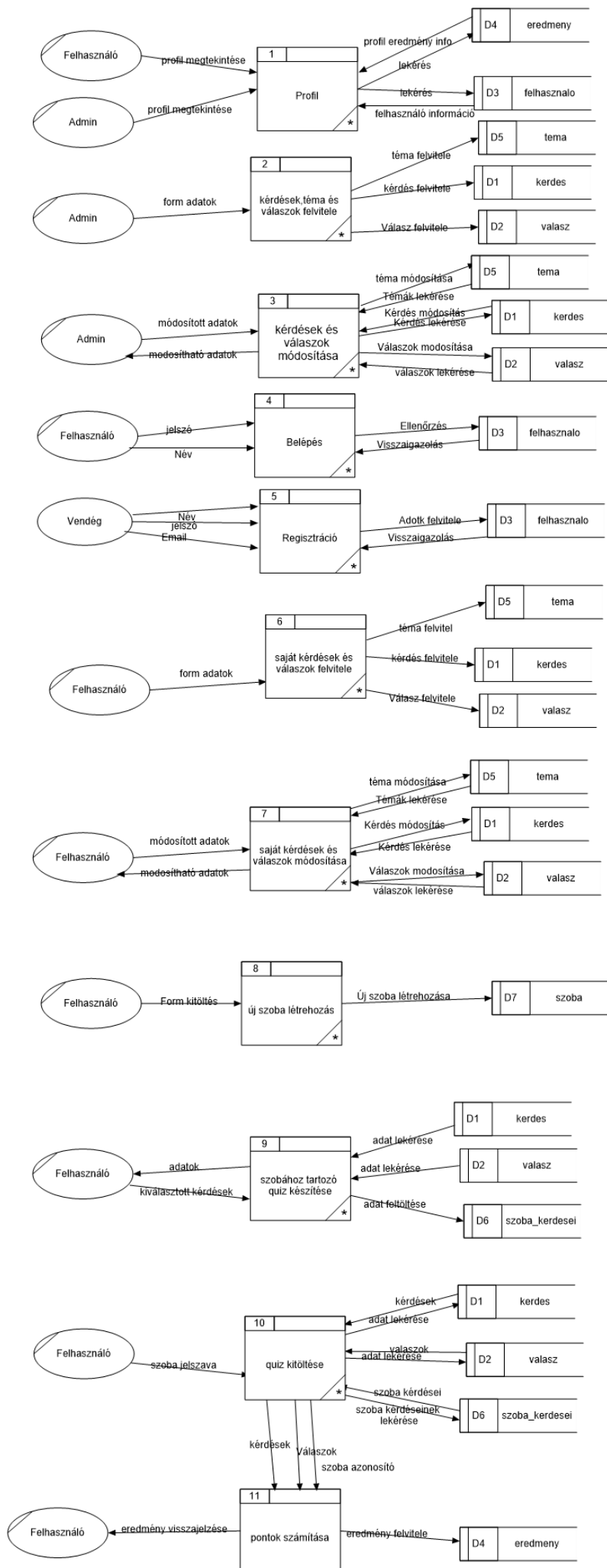
- A játékosoknak lehetősége kell, hogy legyen a kérdésekre való válaszadásra, amelyek véletlenszerűen kerülnek megjelenítésre.
- A játékosok pontokat kell, hogy szerezzenek a helyes válaszokért, és az adott időn belül kell válaszolniuk a kérdésekre.

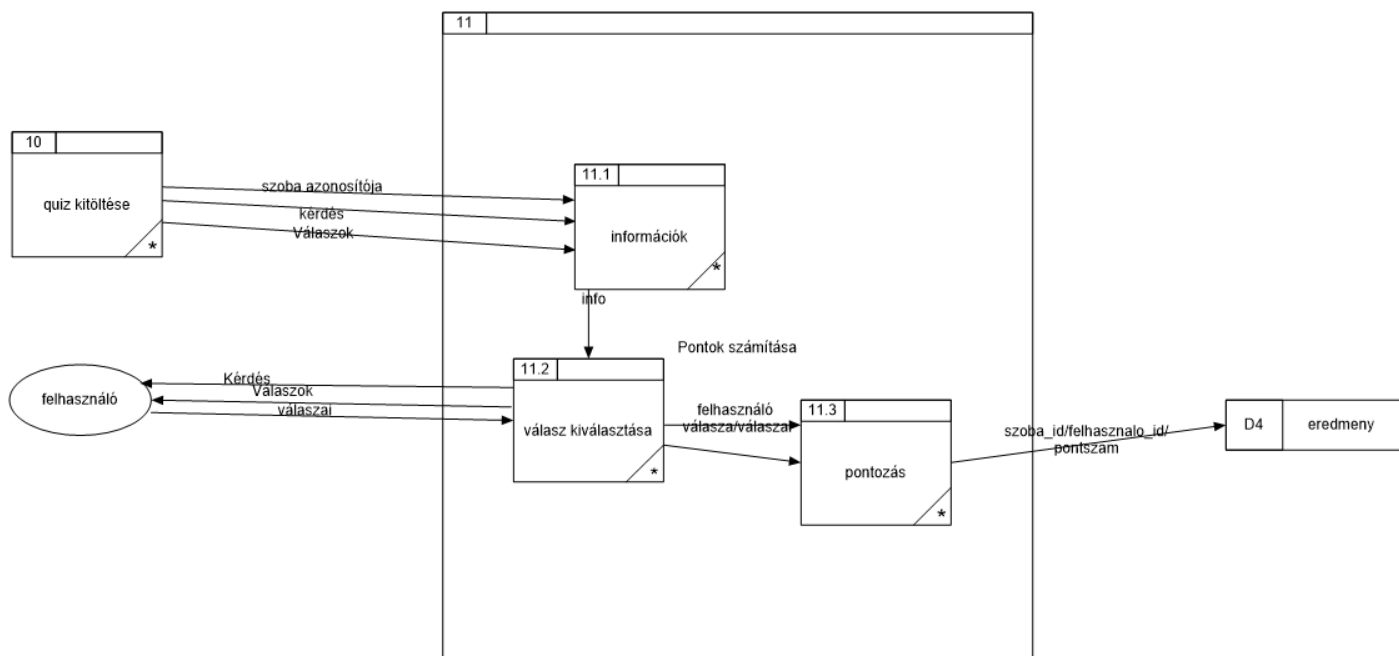
Nem funkcionális követelmények:

- Megbízhatóság
- Teljesítmény
- Felhasználói felület
- A programnak grafikus felhasználói felületet kell biztosítania a felhasználók számára.
- A felhasználói felületnek ergonomikusnak kell lennie, és az információkat jól átlátható módon kell megjelenítenie.
- A felhasználói felületnek tartalmaznia kell a szükséges gombokat és elemeket, amelyekkel a felhasználók könnyen használhatják a programot.
- Karbantarthatóság
- Biztonság
- A programnak biztosítania kell az adatok biztonságos tárolását és kezelését.
- A programnak lehetővé kell tennie a felhasználók jogosultságainak kezelését, például az adminisztrátori jogosultságokat.
- A programnak megfelelő védelmet kell biztosítania a bizalmas adatokkal szemben, például a jelszavak védelmével.

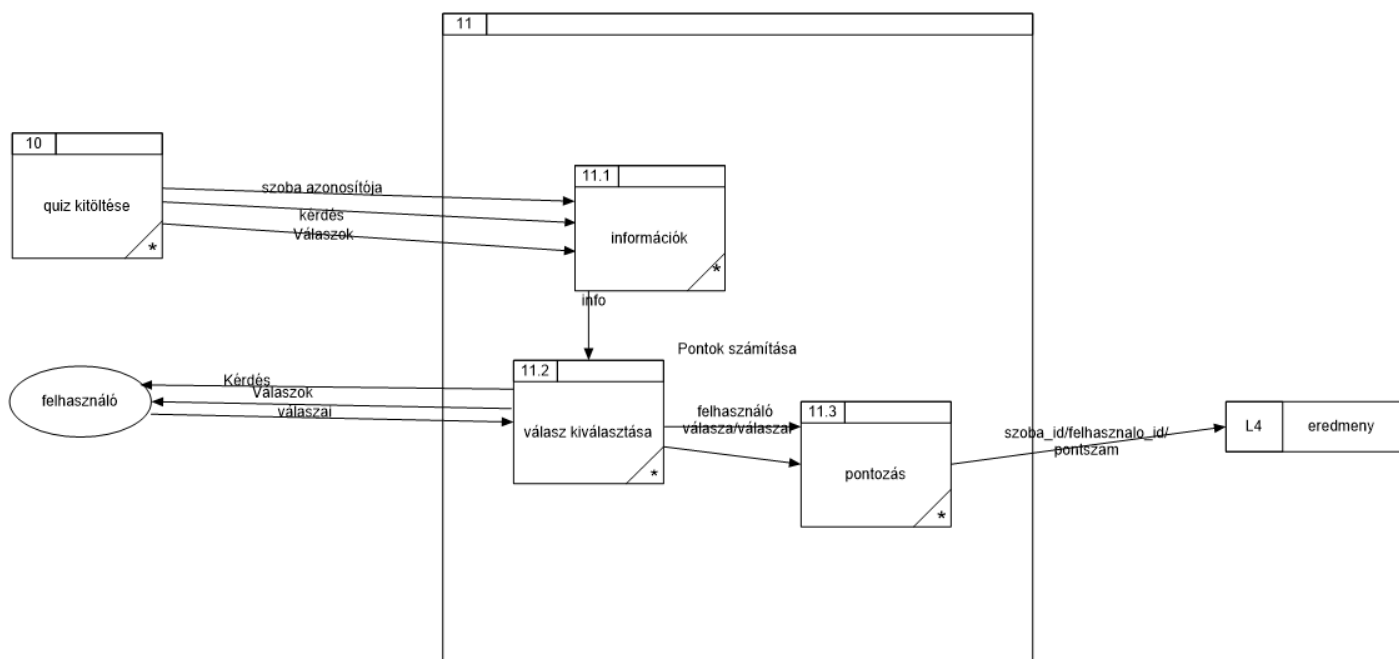
Adatfolyam diagram (DFD):

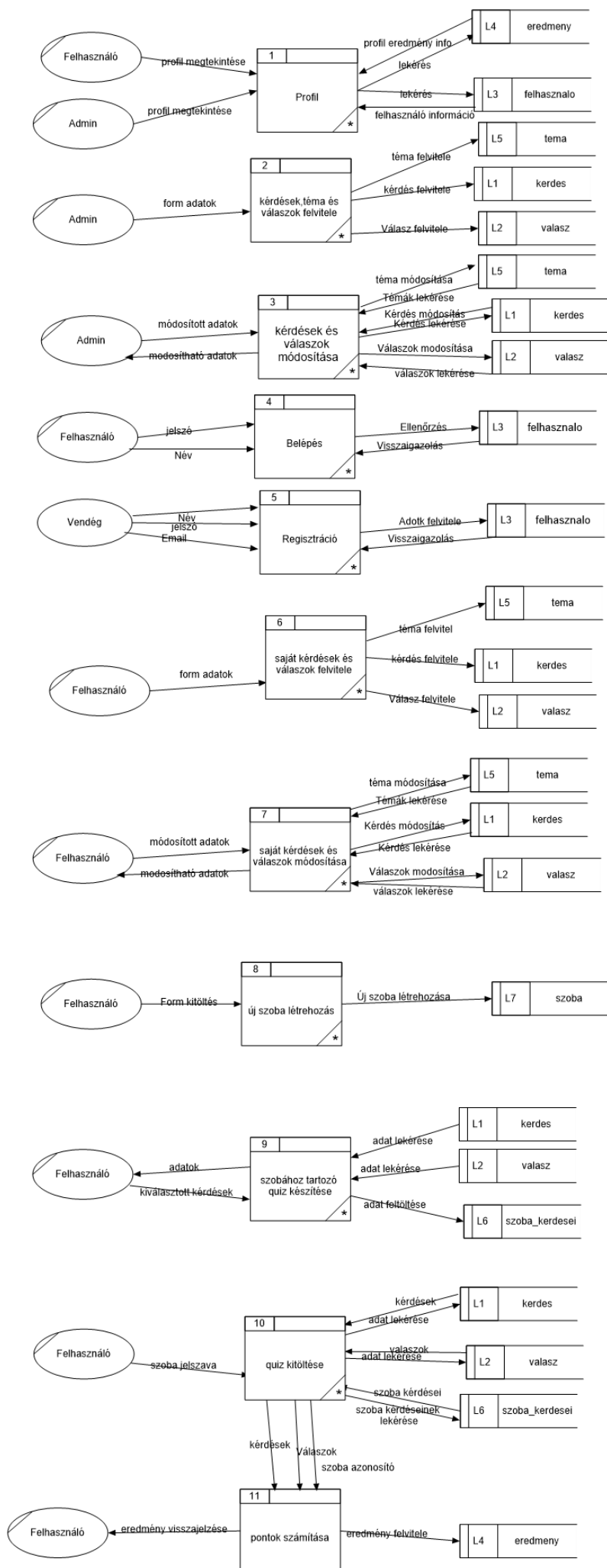
Fizikai



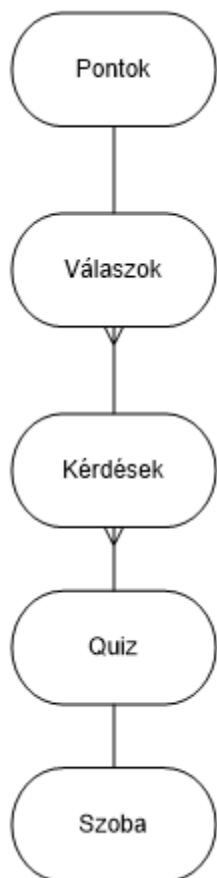


Logikai

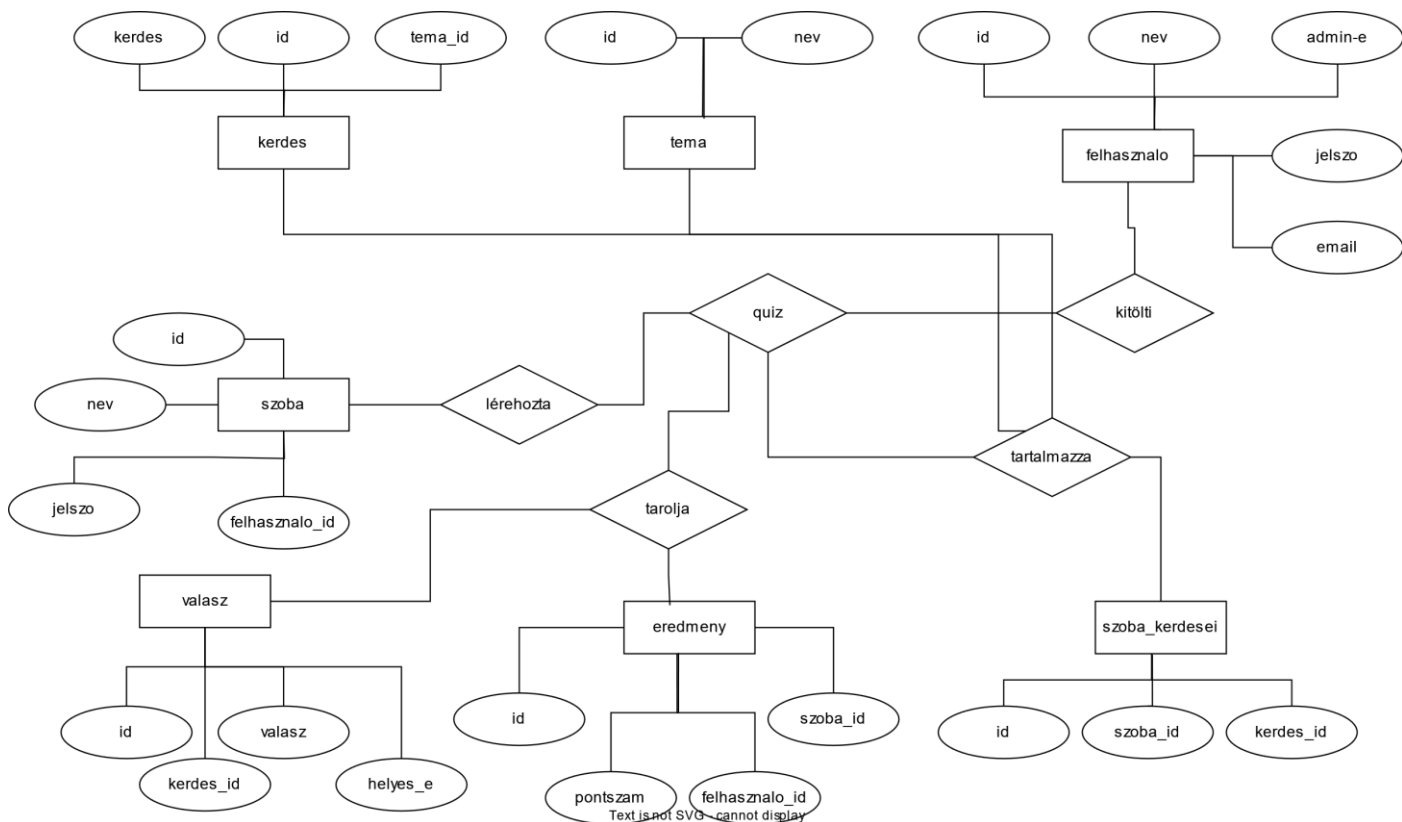




Egyedmodell:



EKT-diagram:



Egyed-kapcsolat diagram leképezése relációs adatbázissémákra:

felhasznalo (id, nev, email, jelszo, admin-e)eredmeny (id, szoba_id, felhasznalo_id, pontszam)kerdes (id, kerdes, *tema_id*)

tema (id, nev)

valasz (id, *kerdes_id*, valasz ,helyes_e)

szoba (id, nev, jelszo, *felhasznalo_id*)

szoba_kerdesei(id, szoba_id, keres_id)

eredmenyei(felhasznalo.id ,eredmeny.felhasznalo id)

szobája(felhasznalo.id , szoba.felhasznalo id)

kerdesei(kerdes.id , szoba kerdesei.kerdes id)

Relációs adatelemzés

1. normálforma (1NF):

- felhasználó: (id, név, email, jelszó, admin-e)
- eredmény: (id, szoba_id, felhasználó_id, pontszám)
- kérdés: (id, kérdés, téma_id)
- téma: (id, név)
- válasz: (id, kérdés_id, válasz, helyes_e)
- szoba: (id, név, jelszó, felhasználó_id)
- szoba_kérdései: (id, szoba_id, kérdés_id)

Megjegyzés: Az adatbázis már eleve kiindulópontjában megfelel az 1NF-nek.

2. normálforma (2NF):

- felhasználó: (id, nev, email, jelszo, admin-e)
- eredmény: (id, szoba_id, felhasználó_id, pontszám)
- kérdés: (id, kérdés, tema_id)
- tema: (id, nev)
- válasz: (id, kérdés_id, válasz ,helyes_e)
- szoba: (id, nev, jelszo, felhasználó_id)
- szoba_kérdesei: (id, szoba_id, kérdés_id)

Megjegyzés: Az adatbázis már eleve kiindulópontjában megfelel az 2NF-nek, hiszen minden táblának van egyetlen elsődleges kulcsa, és nincsenek benne összetett attribútumok.

3. normálforma (3NF):

- felhasználó: (id, nev, email, jelszo, admin-e)
- eredmény: (id, szoba_id, felhasználó_id, pontszám)
- kérdés: (id, kérdés, tema_id)
- tema: (id, nev)
- válasz: (id, kérdés_id, válasz ,helyes_e)
- szoba: (id, nev, jelszo, felhasználó_id)
- szoba_kérdesei: (id, szoba_id, kérdés_id)

Megjegyzés: Az adatbázis már eleve kiindulópontjában megfelel az 3NF-nek, hiszen az egyes táblákban található attribútumok között nincs tranzitív függőség.

Táblák leírása:

felhasználó: *felhasználó tartalmának leírása.*

Név	Típus	Leírás
id	NUMBER	egyedi azonosító
nev	CHAR	felhasználó belépési neve
email	CHAR	Email- cím
jelszo	CHAR	jelszó
admin-e	INTEGER	adminisztrátor-e vagy sem

eredmény: *Ez az adattábla tárolja a játékos eredményét egy adott szobában*

Név	Típus	Leírás
id	NUMBER	egyedi azonosító
szoba_id	FOREIGN KEY	Szoba azonosítója
felhasználó_id	FOREIGN KEY	Felhasználó azonosítója
pontszám	INTEGER	pontszámuk

kerdes: Ez az adattábla tárolja az összes kérdést, amelyek az alkalmazásban megjelennek.

Név	Típus	Leírás
id	NUMBER	egyedi azonosító
kerdes	CHAR	kérdés szövege
tema_id	FOREIGN KEY	Téma azonosítója

tema: Ez az adattábla tárolja az összes témát, amelyek az alkalmazásban megjelennek.

Név	Típus	Leírás
id	NUMBER	egyedi azonosító
nev	CHAR	Téma megnevezése

valasz : Ez az adattábla tartalmazza az összes választ, amelyeket a felhasználók az egyes kérdésekre adhatnak

Név	Típus	Leírás
id	NUMBER	egyedi azonosító
kerdes_id	FOREIGN KEY	válaszhoz tartozó kérdés azonosítója
valasz	CHAR	válasz szövege
helyes_e	INTEGER	A kérdése vonatkozó helyesség

szoba: Ez az adattábla tartalmazza a szoba információját

Név	Típus	Leírás
id	NUMBER	egyedi azonosító
nev	CHAR	szoba neve
jelszo	CHAR	jelszó
felhasznalo_id	FOREIGN KEY	Azon felhasználó azonosítója, aki a szobát létrehozta

szoba_kerdesei : Ez az adattábla tartalmazza az adott szobához tartozó kérdéseket

Név	Típus	Leírás
id	NUMBER	egyedi azonosító
szoba_id	FOREIGN KEY	Szoba azonosítója
kerdes_id	FOREIGN KEY	válaszhoz tartozó kérdés azonosítója

Szerep-funkció mátrix:

Funkciók /Szerepkörök	Felhaszn áló	Admi n	Vendé g
Regisztráció			x
Belépés	x	x	
Profil	x	x	
saját kérdések és válaszok felvitele	x	x	
saját kérdések és válaszok módosítása	x	x	
új szoba létrehozás	x	x	
szobához tartozó quiz készítése	x	x	
quiz kitöltése	x	x	
kérdések és válaszok módosítása		x	
kérdések,téma és válaszok felvitele		x	

Hozzáadott triggerek:

Trigger: update_last_activity_trigger

Leírás:

Ez a trigger minden alkalommal frissíti a felhasználó tábla utolso_aktivitas_datum mezőjét, amikor egy új rekordot szűrnak be az eredmény táblába, és az új rekordnak van egy felhasználó_id értéke, amely megegyezik a felhasználó tábla egyik rekordjának azonosítójával.

Kód:

```
CREATE OR REPLACE TRIGGER update_last_activity_trigger
BEFORE INSERT ON eredmény
FOR EACH ROW
BEGIN
    UPDATE felhasználó
    SET utolso_aktivitas_datum = CURRENT_TIMESTAMP
    WHERE id = :new.felhasznalo_id;
END;
/
```

Trigger: update_room_last_activity_trigger

Leírás:

Ez a trigger minden alkalommal frissíti a szoba tábla utolso_aktivitas_datum mezőjét, amikor új rekordot szűrnak be, meglévő rekordot frissítenek vagy rekordot törölnek a szoba_kerdesei táblában, és az adott rekordokhoz tartozó szoba_id érték alapján azonosítja a megfelelő szobát.

Kód:

```
CREATE OR REPLACE TRIGGER update_room_last_activity_trigger
AFTER INSERT OR UPDATE OR DELETE ON szoba_kerdesei
FOR EACH ROW
DECLARE
    v_room_id szoba.id%TYPE;
BEGIN
    -- Szoba azonosítójának lekérdezése az érintett sor alapján
    IF INSERTING THEN
        v_room_id := :new.szoba_id;
    ELSIF UPDATING THEN
        v_room_id := :new.szoba_id;
    ELSIF DELETING THEN
        v_room_id := :old.szoba_id;
    END IF;

    -- Szoba utolsó aktivitás dátumának frissítése
    UPDATE szoba
    SET utolso_aktivitas_datum = CURRENT_TIMESTAMP
    WHERE id = v_room_id;
END;
/
```

Hozzáadott eljárások:

Eljárás: get_average_score

Leírás:

Ez az eljárás egy felhasználó által elért átlagos pontszám lekérdezését végzi a megadott felhasználó azonosítója alapján. Az eredményt az avg_score kimeneti paraméterben adja vissza.

Kód:

```
CREATE OR REPLACE PROCEDURE get_average_score(user_id IN NUMBER,avg_score
OUT NUMBER)
IS
BEGIN
    SELECT AVG(pontszam)
    INTO avg_score
    FROM eredmeny
    WHERE felhasznalo_id = user_id;
END;
/
```

Eljárás: get_all_room_results

Leírás:

Ez az eljárás az összes szobához tartozó eredményeket kérdezi le.

Kód:

```
CREATE OR REPLACE PROCEDURE get_all_room_results (
```

```

        result OUT SYS_REFCURSOR
    )
    IS
    BEGIN
        OPEN result FOR
            SELECT e.szoba_id, e.felhasznalo_id, e.pontszam
            FROM eredmeny e;
        END;
    /

```

Eljárás: get_toplist_for_room

Leírás:

Ez az eljárás az adott szobához tartozó legjobb eredményeket kéri le. A felhasználó nevét és a pontszámot adja vissza egy kimeneti kurzoron keresztül, rendezve a pontszámok csökkenő sorrendjében.

Kód:

```

CREATE OR REPLACE PROCEDURE get_toplist_for_room (
    room_id IN NUMBER,
    result OUT SYS_REFCURSOR
)
IS
BEGIN
    OPEN result FOR
        SELECT f.nev AS felhasznalonev, e.pontszam
        FROM eredmeny e
        JOIN felhasznalo f ON e.felhasznalo_id = f.id
        WHERE e.szoba_id = room_id
        ORDER BY e.pontszam DESC;
    END;
    /

```

Összetett lekérdezések:

Quiz.sql 137.sor - 142.sor

```

SELECT f.nev AS felhasznalonev, e.pontszam
FROM eredmeny e
JOIN felhasznalo f ON e.felhasznalo_id = f.id
WHERE e.szoba_id = room_id
ORDER BY e.pontszam DESC;

```

quiz_toplist.php 40.sor - 44.sor

```

SELECT s.nev AS szoba_neve
FROM szoba s
JOIN eredmeny e ON s.id = e.szoba_id
WHERE e.felhasznalo_id = :user_id AND s.id = :room_id
ORDER BY e.id DESC

```

new_question.php 77.sor

```
SELECT k.id AS question_id, k.kerdes AS question_text, v.id  
AS answer_id, v.valasz AS answer_text, v.helyes_e AS  
is_correct, k.globalis_kerdes AS is_global FROM kerdes k  
LEFT JOIN valasz v ON k.id = v.kerdes_id
```

quiz.php 85.sor - 89.sor

```
SELECT k.*, v.*  
FROM szoba_kerdesei sk  
JOIN kerdes k ON sk.kerdes_id = k.id  
JOIN valasz v ON k.id = v.kerdes_id  
WHERE sk.szoba_id = :room_id
```